

University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information
Sciences

1-1-2009

Formal concept analysis for an e-learning semantic web

Ghassan Beydoun

University of Wollongong, beydoun@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Beydoun, Ghassan: Formal concept analysis for an e-learning semantic web 2009, 10952-10961.
<https://ro.uow.edu.au/infopapers/1438>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Formal concept analysis for an e-learning semantic web

Abstract

We capture student interactions in an e-learning community to construct a semantic web (SW) to create a collective meta-knowledge structure guiding students as they search the existing knowledge corpus. We use formal concept analysis (FCA) as a knowledge acquisition tool to process the students virtual surfing trails to express and exploit the dependencies between web-pages to yield subsequent and more effective focused search results. We mirror social navigation and bypass the cumbersome manual annotation of webpages and at the same time paralleling social navigation for knowledge.

We present our system KAPUST2 (Keeper and Processor of User Surfing Trails) which constructs from captured students trails a conceptual lattice guiding student queries. We use KAPUST as an e-learning software for an undergraduate class over two semesters. We show how the lattice evolved over the two semesters, improving its performance by exploring the relationship between 'kinds' of research assignments and the e-learning semantic web development. Course instructors monitored the evolution of the lattice with interesting positive pedagogical consequences. These are reported as well in this paper.

Disciplines

Physical Sciences and Mathematics

Publication Details

Beydoun, G. (2009). Formal concept analysis for an e-learning semantic web. *Expert Systems with Applications*, 36 (8), 10952-10961.

Formal Concept Analysis for an E-learning Semantic Web

Ghassan Beydoun¹

¹School of Information Systems and Technology, University of Wollongong, Australia
beydoun@uow.edu.au

Abstract. We capture student interactions in an e-learning community to construct a Semantic Web (SW) to create a collective meta-knowledge structure guiding students as they search the existing knowledge corpus. We use Formal Concept Analysis (FCA) as a knowledge acquisition tool to process the students virtual surfing trails to express and exploit the dependencies between web-pages to yield subsequent and more effective focused search results. We mirror social navigation and bypass the cumbersome manual annotation of webpages and at the same time paralleling social navigation for knowledge.

We present our system KAPUST2 (*Keeper And Processor of User Surfing Trails*) which constructs from captured students trails a conceptual lattice guiding student queries. We use KAPUST as an e-learning software for an undergraduate class over two semesters. We show how the lattice evolved over the two semesters, improving its performance by exploring the relationship between ‘kinds’ of research assignments and the e-learning semantic web development. Course instructors monitored the evolution of the lattice with interesting positive pedagogical consequences. These are reported as well in this paper.

1. Introduction work

Building a semantic web (SW) is often hindered by these two factors: users do not agree on an annotation standard that can be used to extricate web inter-dependencies and/or users feel burdened by the annotation effort. We proposed an alternative in (Beydoun, Kultchitsky et al. 2007), based on indirect social navigation (Dieberger 1997), to exploit web-pages dependencies using surfing trails left behind. These are not beneficial if individuals surfing the net have different interests, but in a given interest group individuals produce trails that are of interest to the whole group. Our early experiments in e-learning in (Beydoun, Kultchitsky et al. 2007) indicated processing surfing trails left by students using Formal Concept Analysis (FCA) is possible. But lack of crossings between surfing trails lowered the usability of the resultant lattice. Our hypothesis in this paper is that the effectiveness of the resultant conceptual lattice depends on a sufficient complexity of the conceptual lattice itself on one hand and the following factors on the other: the way assignments are set to ensure trails crossings, a reflective learning setting, and a sufficient number of trails. In this paper, we employ our FCA based system for two consecutive semesters in an American University and using comparative assignments to ensure higher number of trail crossings. The conceptual lattice resultant from the first semester is used as a starting point for further collective development by students in the second semester. The number of crossings is much higher and the experiments confirm that indeed social navigation can be simulated in e-learning settings, and moreover that FCA is suitable to process the trails into a usable semantic web which supports the learning process of students.

The paper is organized as follows: Section 2 overview related work, Section 3 presents the conceptual basis of our FCA approach in e-learning, Section 4 present our system KAPUST, Section 5 presents our experimental results, Section 6 discusses and analyses these results from both perspectives, technical and pedagogical concluding with a discussion of future work.

2. Related work

To effectively use web-surfing experience of people, a user friendly exposure of the experience that all people can understand is required. In (Beydoun, Kultchitsky et al. 2007), we advocated building a SW incrementally by the users themselves, where no intermediate expert is needed, and ontologies are not predetermined. In capturing and organizing user trails according to the browsing topic, and later allowing

intelligent search of the web, users provide their topic of interest, within their interest group, and begin browsing web pages. Submitted topics of interest and trails left behind by users form the raw information to constitute the SW structure, and determine how it evolves. We process this raw information using machine learning. Our approach expanded the work of Wexelblat and Maes at the MIT Media Lab, which highlighted the importance of tracking user trails and interactions to benefit from interaction history for information navigation (Wexelblat 1999). We applied Formal Concept Analysis (FCA) (Ganter and Wille 1999) to reason about the traces instead of only displaying and browsing trails as in (Wexelblat 1999). The reasoning component has a retrieval knowledge base (the actual SW) which integrates users knowledge scattered in their left-behind surfing traces. Our FCA-based approach imposes very little effort on users and bypasses manual marking up of current tools to build the SW. Our system, KAPUST, captures users behaviors and stores them for analysis and reasoning. In capturing user traces, it is similar to (Laus 2001) and to (Wexelblat 1999; Wexelblat and Maes 1999) which store interactions history on a user basis. The kind of information of interest to us in this paper is the internet parallel of *social navigation*, which is human interactions in pursuit of information gathering (Dieberger 1997).

Formal concept analysis has been used in various classification tasks e.g. to classify software engineering activities (Tilley, Cole et al. 2005) and to impose structure on semi-structured texts e.g. (Kim and Compton 2001; Eklund and Wormuth 2005). It has been found efficient when applied to document retrieval systems for domain specific browsing (Kim and Compton 2001). Our use of FCA is unique in that the lattice is built in collaborative manner reflecting the collective meaning of *user trails*, and modelling *unintended* and *indirect* social navigation over the web: users are not intentionally helping each other (e.g. following footsteps in a forest) and they do not directly communicate. We build a complex information space (the SW), where we analyze traces left by users. Our approach is similar to Footprints (Wexelblat 1999; Wexelblat and Maes 1999), where a theory of interaction history was developed and a series of tools were built to allow navigation of history rich pages and to contextualize web pages during browsing. However, unlike our approach, it does not use history to make recommendations and nor does it have a reasoning technique embedded in it

Creating an on-line collaborative learning environment is a necessary aspect for e-learning which is the systematic use of networked information and communications technology in teaching and learning (Horton and Horton 2003). E-Learning is flexible, relatively cheap and supplies “just in time” learning opportunities. E-learning is directly underpinned by the development of and access to information and communications technology infrastructure. Creating a sense of community and understanding the on-line behaviors of the participants are also crucial (Blunt and Ahearn 2000). Distant academic learning is one important application. E-learning techniques in the corporate world are also often used for residential workshops and staff training programs. Several efforts have been made to create e-learning environments. Notably, in George Mason University under the Program on Social and Organizational Learning (PSOL), research is being done to create and maintain a Virtual Learning Community for the participants in the program. The purpose of that research is studying the learning of the community within the developed environment and a better understanding of the dynamics of collaborative dialogue to enable more informed and sound decision making (Blunt and Ahearn 2000).

3. Constructing the Semantic Web with User Trails and Formal Concept Analysis

In our Semantic Web, search ontologies are evolved from the keywords that users enter to name their browsing sessions at the start. These keywords relate to users search domains. How these naming keywords are transformed into a semantic web is described this section.

A group of students sharing an assignment problem usually discuss the assignment topic meeting face to face every day at university. Our approach creates an interactive environment to simulate those discussions and turn the collective knowledge generated from such discussions into a comprehensible semantic web that can be accessed not only by the students of the current class, but also students who will take the same class in following semesters are able to access it and develop it further. The web is a conceptual lattice developed using Formal Concept Analysis which is a machine learning technique. In our approach, individual traces are the data points that the FCA algorithm uses for learning. Traces are stored as a sequence of URL of pages that users visit in a *browsing session* when a user from particular interest group is searching for a specialized topic. For example, in e-learning, users are students who provide one or more keywords to identify their search domain at the beginning of each session (Figure 1). Their trails consist of sequence of URL annotated by the session title word(s) entered at the beginning of each web session. Web page addresses and session title keywords are the building blocks for our SW. Initially

entered words are checked against dictionary of existing set of keywords in the database. This minimizes the redundancy of keywords (e.g. synonyms) and corrects any syntactical errors by users. The evolved SW structure gives authenticated users recommendations in the form of categorized web page links, based on session keywords. In addition, they can browse any notes added previously by authorized users. As user trails are accumulated, browsing sessions begin to intersect one another to create new knowledge. For example, while a student *A* searches web course notes for pages about the “*Public Sector*”, she comes across a webpage *p1*, which has been visited by student *X* but it was related to “*IT, E-Learning*” in his session. This creates new knowledge relating “*IT, E-Learning*” and “*Public Sector*” due to intersection in the corresponding trails.

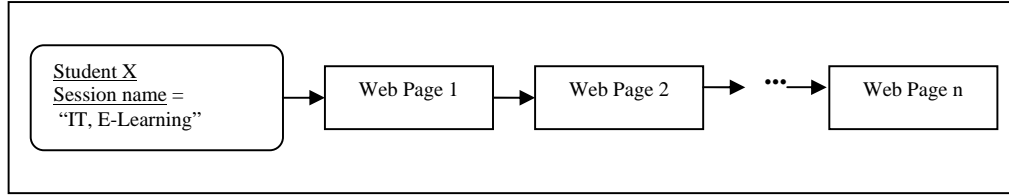


Figure 1 User Trail: A student *X* searching for webpages related to *IT* and *E-Learning*: she logs in, enters session title, and browses for related articles. Web pages 1 to *n* will be given under any session with title keywords *IT* and *E-Learning*.

FCA (Ganter and Wille 1999) reasoning turns user traces into structured knowledge, the conceptual lattice. This involves two steps: a matrix table is constructed showing keywords that each page satisfies, a conceptual lattice is then assembled from the matrix table. FCA starts with a context $K = (G; M; I)$, where G is a set whose elements are called objects, M is a set whose elements are called attributes, and I is a binary relation between G and M [$(g;m) \in I$ is read “*object g has attribute m*”]. Formalized concepts reflect a relation between objects and attributes. A *formal concept*, C , of a *formal context* $(G; M; I)$ is a pair (A, B) where $A \subseteq G$ is the set of objects (extent of C) and $B \subseteq M$ is the set of attributes (intent of C). The set of all formal concepts of a context K together with the order relation \leq is a *complete lattice*, $F(G, M, I)$. For each subset of concepts there is always a unique greatest common subconcept and a unique least common superconcept. In our approach, web page URLs form G , the set of objects. Keywords of session names form M , the set of attributes. A concept in the resulting conceptual lattice is formed of a set of page URLs as the extents and a set of keywords as the intents. Concepts can be a result of either a single user session or multiple sessions that intersect each other.

Figure 2 displays an example of 3 different user-sessions that share some common web pages in their trails. For example, “*Web Page1*” is visited by users *A* and *C* having different keywords identifying their session. This indicates the creation of a new concept in the lattice as a result of the intersection between their sessions. The new concept will have “*WebPage1*” as its page set and “*IT, Political Science, Technology*” as its keyword set. The concept having “*WebPage 1*”, “*WebPage 2*”, “*WebPage 3*” as its page set and “*IT*” as its keyword set, is an example of a concept resulting from a single user, *A*, session.

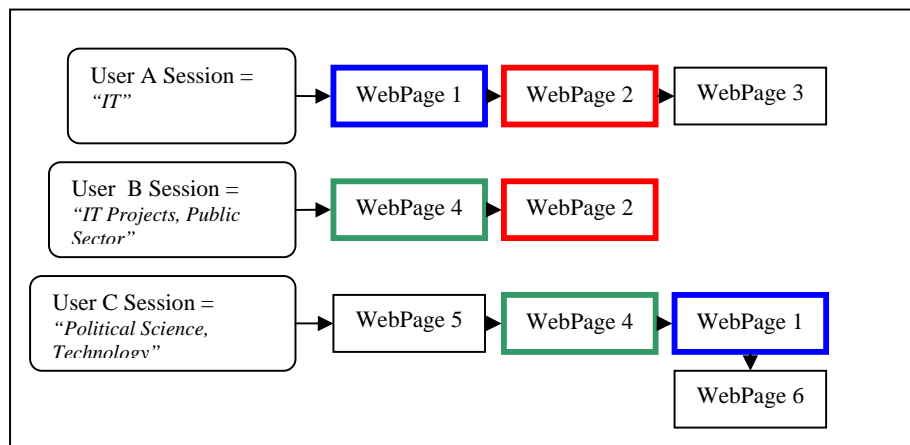


Figure 2 : Example of user trails

Sets of names and web page URLs are extracted from user traces and input to the FCA engine as XML documents. Traces are stored in a database together with the relations that exist between each web page URL and its session name. At this stage the input to the FCA engine contains all user traces collected so far. As more traces are collected, they are incrementally fed to the FCA engine to update the existing matrix table with any new web page URLs and keywords (Figure 3).

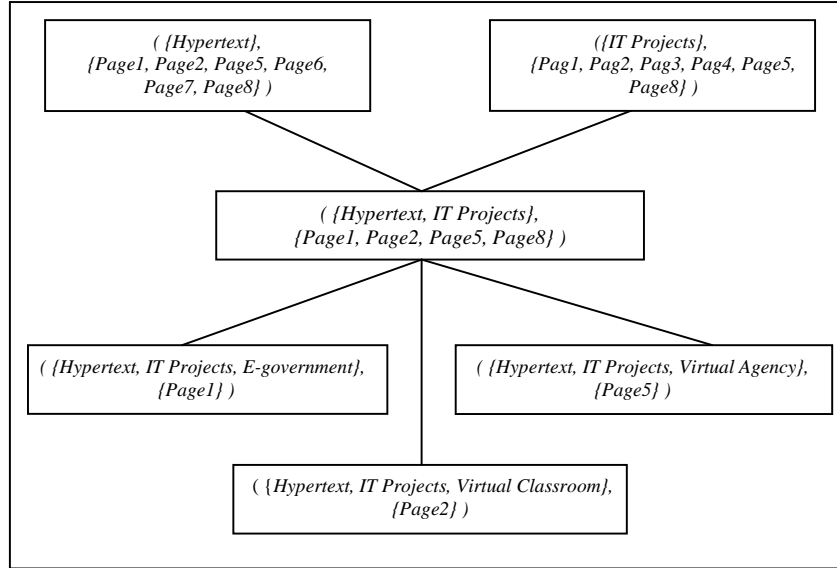


Figure 3 conceptual lattice: An E-learning example

With every new matrix table and a set of new traces, the FCA engine reconstructs the conceptual lattice. For our learning application, the lattice is updated on weekly basis following each class. The lattice generation can be scheduled to run daily instead of weekly in case of higher usage of KAPUST. KAPUST subsequent use of the generated lattice for query management, and intelligent interface are described in the next section. Our querying algorithm takes a user's query (set of keywords) and the conceptual lattice as input. It returns as output, the web page links that best match the search query (Figures 3 and 4).

A lattice, L , is a tuple of two sets, (P_i, K_i) , where P_i and K_i as sets of page links and keywords respectively (Figure 3). To illustrate query processing by KAPUST, we denote the set of potential concepts that match the user request, together with their priorities as $PotentialConcepts = \{(P_i, K_i), Priority\}$, where $Priority$ determines how relevant the concept is to the user's query. We take it as the depth level of a concept (P_i, K_i) in the conceptual lattice in case a matching concept is found. Otherwise, we take it as a measure of how many keywords from the set of keywords entered by the user at login, UK , exists in a concept (P_i, K_i) . Referring to Figure 3.4, the querying algorithm has the following steps:

Step 2 in the process prunes the set of keywords entered by the user by removing new keywords that do not exist in the concept lattice.

Step 3 checks for a concept that has the exact set of pruned keywords.

Step 4 handles the case where no matching concept is found. In this case, all concepts that have one or more keyword in their set of keywords that matches any keyword in UK' are added as potential concepts. The priority is taken as $(CountUK - CountK)$ to give highest priority to the concepts that have more matching keywords. If a concept has 2 matching keywords and the set UK' has 3 keywords in its list, the priority will be 1, which is higher than a concept that has 1 matching keyword where the priority will be 2.

Steps 5 and 6 consider the case where a matching concept is found. They add super concepts and/or sub concepts of the matching concept. Sub concepts will have a higher priority than the super concepts because they're more specialized. The most general and most specific concepts are not considered as super or sub concepts. If no super or sub concepts are found, the matching concept itself is added to the potential list of concepts.

Step 7 orders the *potentialconcepts* for display.

Step 8 divides the category and the page links under each category then retrieved the average rating for each page link to be displayed for the user.

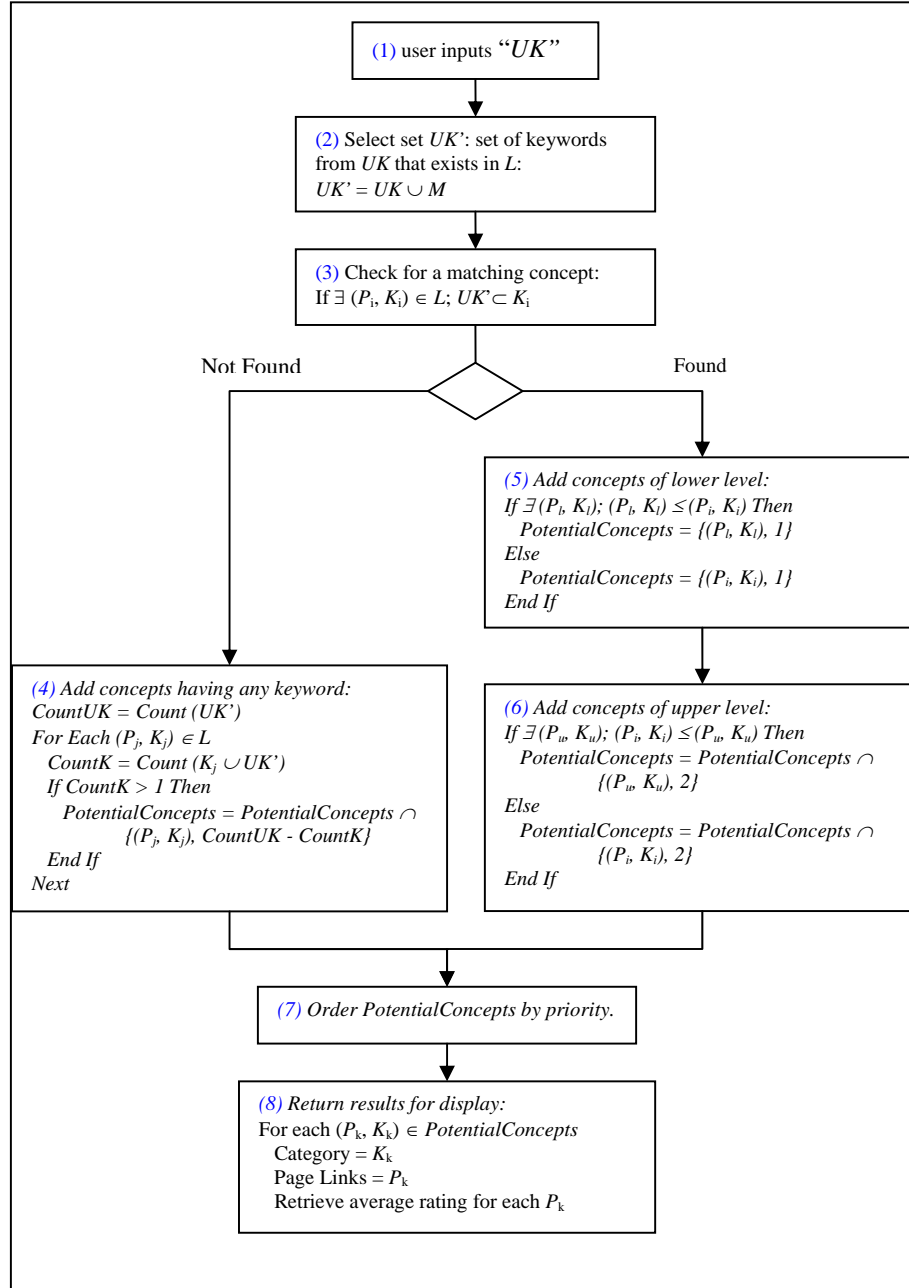


Figure 4. Processing Queries Using the Lattice

The strategy of choosing super and sub concepts gives the user a better perception and a wider amount of relevant information. Sub concepts contain all extents of the concept itself. This allows us to categorize the page-sets one level deeper. Back to our example in Figure 4, suppose the student logs in and enters keywords “*Hypertext, IT projects*”, there is a matching concept containing exactly those keywords. Instead of getting the following results:

‘*Hypertext, IT projects* → WebPage 1, WebPage 2, WebPage 5, WebPage 8’

KAPUST gives the student a better insight about those webpages with the following recommendations:

Hypertext, IT Projects, E-government: WebPage1
Hypertext, IT Projects, Virtual Classroom: WebPage2
*Hypertext, IT Projects, Virtual Agency:*WebPage5
Hypertext: WebPage 6, WebPage 7, WebPage 8
IT Projects: WebPage 3, WebPage 4, WebPage 8

Degree of generality of a concept on a page and its priority are inversely related. Recommended pages are shown to the user in order of decreasing priority. A page is displayed once unless, it belongs to n concepts on the same level of generality in the lattice, it will be displayed n times. Search results given by KAPUST is shown in Figure 5.

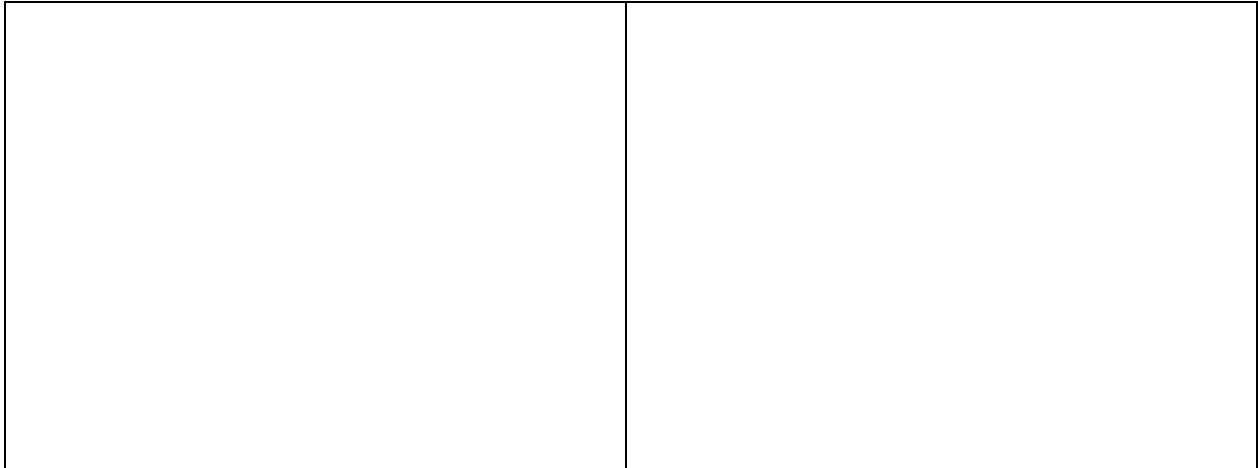


Figure 5 Screen shots for the annotation tool: **Login module (left frame)** a user has just logged in to start a new browsing session about “IT” (right frame) **Right** KAPUST search results are lists of rated and categorized links, to visit any page, users may click on their links benefiting from traces of others their community.

4. Architecture Keeper and Processor of Users Surfing Trails (KAPUST)

In this section, we present KAPUST, its underlying sub systems and the deployment scenarios that have been arranged to use in an E-Learning environment at the American University of Beirut. KAPUST architecture (Figure 6) has two components: an extensive interactive user interface (See Figure 5) and a reasoning/knowledge creating part (invisible to the user). The reasoning components implements the conceptual framework discussed in the previous section. KAPUST is a distributed intelligent system. Its deployment requires individual deployment packages on nodes of the network. This is later described after we describe the user interface.

The interactive component organizes the user input to the machine learner, and interfaces the user with existing knowledge. KAPUST search results are lists of rated and categorized links, to visit any page, users may click on their links benefiting from traces of others their community (See Figure 5 right). In our e-learning example (detailed later), students get to exploit views of each other. They might find their quests in those links, or they can go ahead and continue searching through other pages and contribute to the traces database for the benefits of future users. Lastly, the user interface checks the session naming, minimizing syntactic mistakes in session keywords and detecting synonyms and homonyms. Similar sounding characters (e.g. ‘C’ and ‘S’) have same code. Another function compares the difference of the SOUNDEX pattern results and returns a number representing difference between the 2 strings. We use this function to determine if a similar string exists. For example, applying SOUNDEX on McDonnell gives the code M-235. Suppose a keyword ‘IT projects’ exists in the database in the list of keywords and the user starts a new search session and enters any of ‘IT’, ‘IT-projects’ or even ‘IT-projctcs’. He would get a suggestion telling him that a similar keyword ‘IT projects’ already exists. He can then either use the suggested keyword or ignore it.

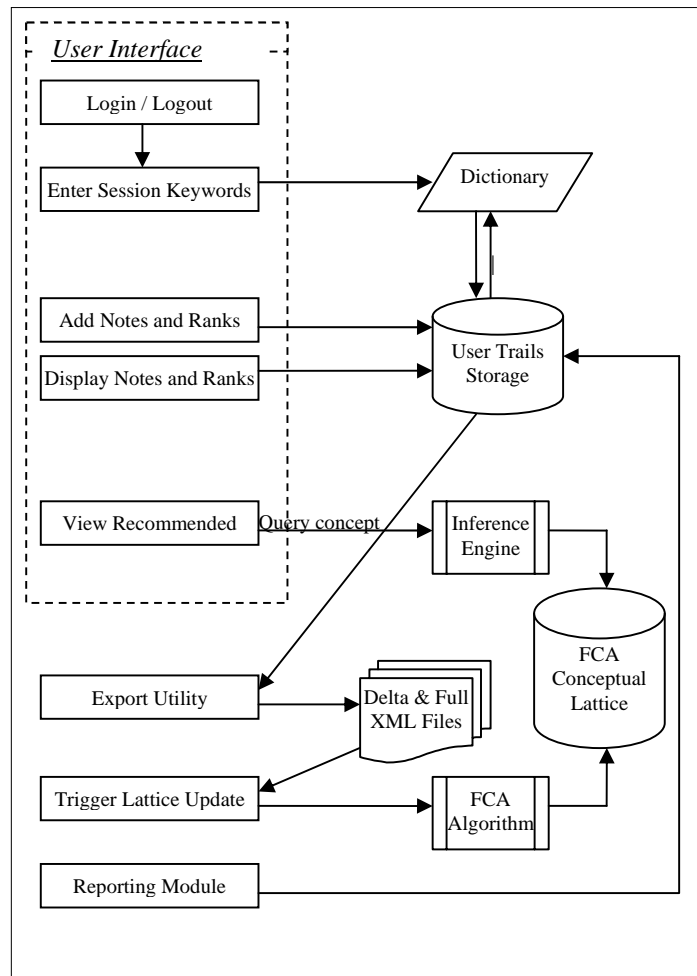


Figure 6 : KAPUST1.2 Architecture

A web server is needed to set up and store the semantic web component of KAPUST. A database server (SQL) is also required to store traces and execute the associated FCA engine. A client machine is designated as *administrator* and accesses the FCA engine from any machine. Traces between clients and KAPUST server are transferred in XML (see Figure 7). In our e-learning environment, students can do their assignments from home. They are given an installation package for the client side of KAPUST. To deploy the application, an IIS Web Server is needed to set up the website that the annotation tool, the export utility and reporting services will be using. An SQL Server is also required to install the database of the annotation tool and the FCA engine. These are needed on the server side. We had two scenarios to set up the client machine that will access the annotation tool. The need and detail of each scenario will be explained next. The administrator's client machine, however, can access the export and reporting utilities as well as call the procedures of the FCA engine from any machine having Internet Explorer and an internet connection.

In our e-learning environment, we needed students to be able to do their assignments either from home, by installing the annotation tool using scenario 1 (Figure 7), or from within the university by accessing the tool using scenario 2 (Figure 7). Due to university policies, students are not allowed to install applications on university computers. For that purpose, a citrix server was set up at the university where the annotation tool was installed. Taking into consideration that most computers at the university have citrix client installed, the users were able to connect through citrix to the server where the annotation tool was installed and use the Internet Explorer browser from that server. Another reason for using citrix server scenario is because the university computers are accessible to all students on its campus. Since we are taking a specific domain for research we don't want to mess up the data with irrelevant information that might result from the improper usage of the tool by other student who are not taking the PSPA course.

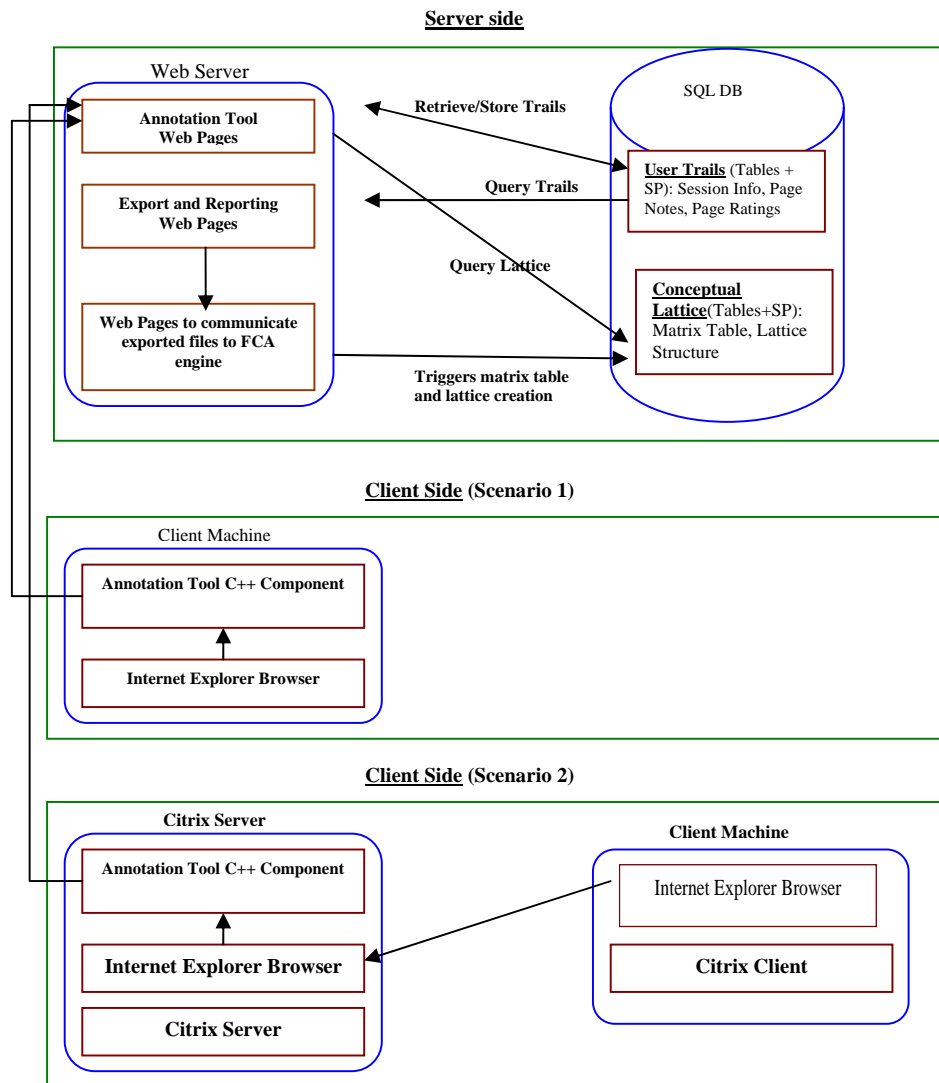


Figure7. KAPUST Deployment/Usage

Students who want to use the tool from home can also use scenario 2 if they install citrix client at their PCs; but that means a slower connection and a longer waiting time compared to scenario 1 where they directly install the annotation tool at their PC. For scenario 1, a package was created that contains the annotation tool installation files. Students only have to run the package to be able to view that annotation tool as an add-on in their Internet Explorer browser and start doing their assignments.

5. E-learning Experimental Results

We deploy KAPUST to provide an E-Learning environment for undergraduate class in the Political Science and Public Administration Department at the American University of Beirut for two consecutive semesters. The course, *Information Technology and Public Administration*, focuses on the impact of IT on various aspects of public administration and policy around the world. The students are senior students from various other majors. They use KAPUST as part of their weekly assignments to answer assigned research questions. The aim of the system in this environment is to give the students a tool to use while doing their assignments to share each Other thoughts as they read relevant pages online. The system's database and

website are deployed on an online server at the university. A setup package to install the internet browser add-on utility, together with a video manual regarding the tool is arranged and distributed to each student so he/she can install and use it from home. Students are free to use KAPUST at home and in some lab sessions to browse for related articles and answer their assignment questions.

For each of the first 12 weeks each semester, the professor gives a research assignment on a new topic. Browsing traces are collected on a weekly basis (as XML exported files (Delta and Full)). KAPUST also allowed the professor to track the students work and keep the collected information for another class to consume. Students are graded for their interaction with the software. They are motivated to use KAPUST as they knew that their colleagues are reading their comments and annotations. Before a student searches for articles to answer the question under study, they need to log in. For each assigned question, students choose one or more keywords from the domain of the assignment question and enter them as session keywords before browsing any web pages. All visited webpages in this session are later annotated with these keywords. Each assignment question is handled as a distinct browsing session by the annotation tool. At a subsequent session login the student chooses a new set of keywords representing a new question.

Semester1 had 12 students. It was seen as an experimental setting and the presence of the professor is maintained throughout each lab. Semester1 results provided an initial conceptual structure for semester 2 when only a lab assistant was present. Students used the conceptual lattice produced by students from Semester 1 as a starting point for their learning experience and continuously added to it. Figure 8 displays information about traces collected in semester 1. Week 1 represents the testing phase where students were getting familiar with the tool. In Semester 1, weeks 2 to 7 were used to collect traces to build the semantic web structure which was built for the first time in week 8 indicating the start of knowledge sharing between students. Week 8 of semester 1 represents the maturity of the semantic web structure and the construction of the first conceptual lattice. During week 9 to 12 of semester 1, the functionality of the system as a whole is observed. In this period, the semantic web structure is continually updated online, the matrix table and the lattice structure are incrementally updated and constructed on a weekly basis, students continue to share their experiences, and most importantly students are given recommendations for each new browsing session they initiate.

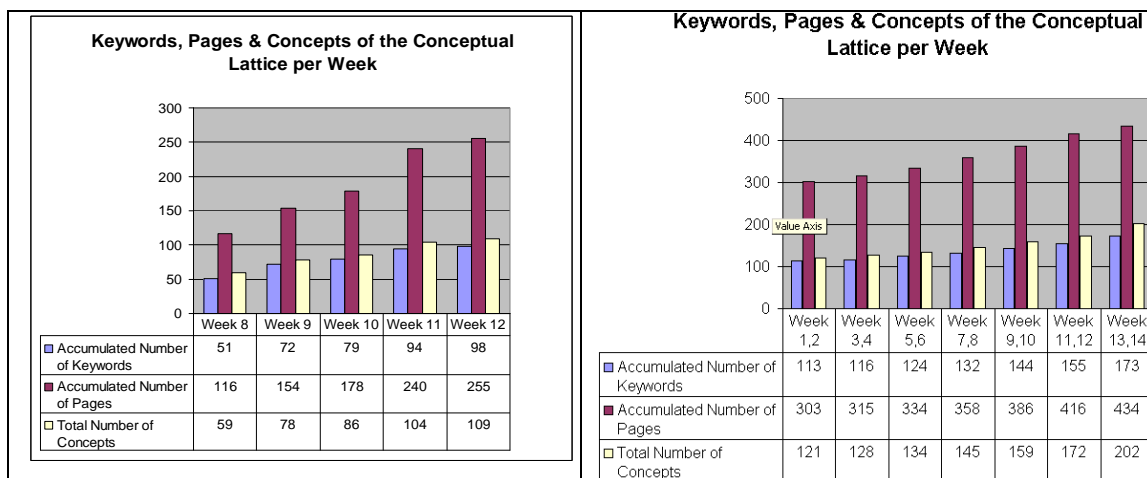


Figure 8 Lattice Evolution: Semester 1 (left): The accumulated number of keywords and pages come form the Matrix Table, while the total number of concepts comes form the Lattice Structure. **Semester 2 (Right)**

The annotation tool successfully collected user traces over the 12 weeks of experimentation (Figure 8 left). A large number of traces were collected. The rate of collecting traces increased over the last few weeks. We attribute this increase to the deployment and use of the lattice structure. Using the conceptual lattice gave the students a kind of motivation to work harder and helped in their learning process. It is encouraging to get relevant links directly to the point one is searching for. Moreover, students started to cross one another's trails after week 7. The first FCA conceptual lattices were generated in week8 (see Figure 9) This shows the power of the inference engine of the tool. The inference engine started taking action during this period to query the conceptual lattice and present recommendations to the students according to their search criteria. The first full exported file is fed to the inference engine in week 8 to

generate the first version of the conceptual lattice. 60 user sessions are collected, 112 notes and 156 ratings are made. Figure 8 (left) shows how the conceptual lattice evolves after the introduction of the lattice by measuring the number of keywords, pages and concepts; concepts are scattered across 8 levels (including most specific and general concepts). The conceptual lattice is finally formed of 255 pages, 98 keywords and 109 concepts. Table 1 displays characteristics of concepts at each of level of the final lattice. Recall that a concept is formed of a set of pages and a set of keywords (characterizing the session), level 1 contains the most general concept (see table 1). This concept is formed of all the pages in its page set and contains the empty set in its keyword set. At level 8 we have the most specialized concept. This concept contains the empty set in its page set and all the keywords in its keyword set.

	# of Concepts		Keywords per concept		# of pages per concept	
	Semester 1	Semester 2	Semester 1	Semester 2	Semester 1	Semester2
Level 1 (General)	1	1	0	0	255	434
Level 2	59	121	1 to 2	1 to 3	1 to 22	1 and 23
Level 3	21	34	2 to 3	2 to 3	1 to 12	1 and 12
Level 4	16	20	3 to 5	3 to 5	1 to 9	1 and 10
Level 5	6	7	5 to 8	5 to 9	1 to 6	1 and 7
Level 6	4	5	8 to 9	7 to 8	1 to 4	1 and 6
Level 7	1	5	12	7 to 9	1	2 and 4
Level 8	1	4	98	9 to 12	0	1 and 3
Level 9	0	3	0	10 to 12	0	1 and 2
Level 10	0	1	0	12	0	1
Level 11 specific	0	1	0	173	0	0

Table 1 Analysis of Conceptual Lattice by Level: the number of keywords per concept increases- in reverse to number of pages, as we move towards the more specialized concepts.

The fact that the lattice has several levels indicates knowledge sharing by the students. It also means that the tool has been successful in mapping concepts with the corresponding pages from different user trails to generate new concepts on various levels of the lattice structure. At level 7, the second most specialized concept, we have 1 concept which refers to the *Google* search engine in its page set and which has 12 satisfied keywords. This concept is not too relevant, because the *Google* search engine is not a valid web page. It's rather a general site. This concept resulted from improper usage of the tool and from testing or noise data during week 1

Semester 2 had 14 students. Observations made in semester 1 led to modifying the style of questions set for semester 2 to ensure more crossings between surfing trails in the second semester. Because the already mature semantic web, it was regenerated using traces collected on a bi-weekly basis (rather than weekly). Table 2 shows the number of pages visited by users per week, as well as the number of crossing pages per week (pages that have been visited by 2 different users). Crossing pages give us a measure of how much sharing of knowledge is occurring. The number crossing pages indicates that the tool has helped students to find their target by using the stored set of previously visited pages. Compared to the results of the earlier semester, once notices that the crossing pages occurred more often and this is due to having an initial lattice structure at the beginning of the semester that further matured during the semester. Recall that last semester, the first 7 weeks were only to build the lattice and it was not until the final weeks that crossing pages started to appear.

	Week 1,2	Week 3,4	Week 5,6	Week 7,8
# of Pages Visited	23	6	24	28
# of Crossing Pages	2	2	0	4
	Week 9,10	Week 11,12	Week 13,14	
# of Pages Visited	35	32	54	
# of Crossing Pages	4	2	32	

Table 2. Traces and crossings during semester 2

Figure 9 shows how the number of concepts evolved on a bi-weekly basis. The accumulated number of keywords and pages come from the corresponding matrix Table, while the total number of concepts comes from the lattice structure itself. The final lattice is scattered among 11 levels and is formed of 434 pages,

173 keywords and 202 concepts. The table shown in Figure 9 displays the characteristics of the concepts at each of these levels for the final lattice structure.

Table 1 shows the most general which is formed of all the pages in its page set and contains the empty set in its keyword set. At level 11 we have the most specialized concept. This concept contains the empty set in its page set and all the keywords in its keyword set. This means that there is no page that satisfies all keywords and there is no keyword that satisfies all pages visited. As in the earlier analysis, notice how the number of keywords per concept increases as we move towards the more specialized concepts, and how the number of pages decreases in that order. This shows that our lattice is appropriately constructed according to the sub-concept and super-concept definition in Formal Concept Analysis. More importantly, the lattice formed in semester 2 was clearly deeper. Semester 1 final lattice had 109 concepts scattered over 8 levels. Semester 2 lattice continued to grow as seen in the above figures and had increased by 3 levels with 93 additional concepts.

The performance of our FCA algorithm in constructing the conceptual lattices is shown in semester 2 Figure 9. It starts at the point where semester 1 left off. The FCA algorithm execution time is minor compared to that needed to collect the traces. In our e-learning domain, traces were collected on weekly basis and the regeneration process was ran every week or bi-weekly basis and as a batch process offline. However, if the system is to be deployed in another domain where lattices are developed many times per hour, then our FCA algorithm may need to be reconsidered.

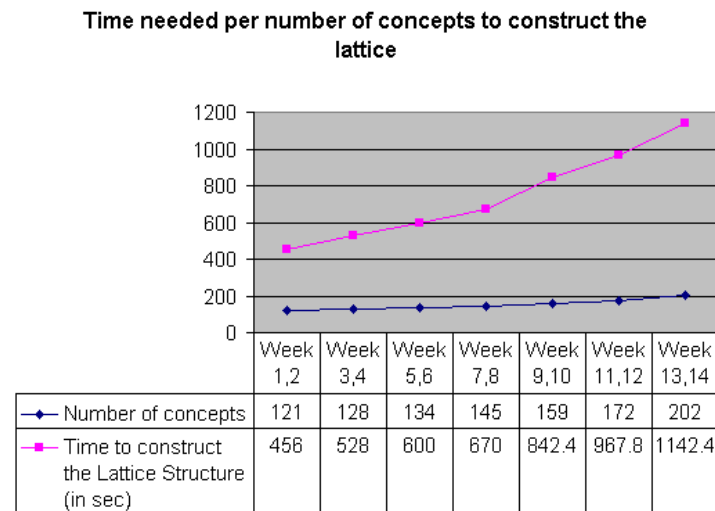


Figure 9. maturity of the lattice and the time required for its regeneration

6. Discussion of results

Collected results are evidence for suitability of reasoning over user traces using KAPUST. Our FCA algorithm used to construct the matrix table and lattice performed well in our domain. The method we use to query the lattice provides recommendations for the students in a categorized way and that gave the students a way to share their knowledge with their fellow students. Querying for a set of naming keywords where none of the concepts in the lattice structure contains an exact match, we take each keyword individually and leave it to the user to judge, this is like an ordinary keyword search thus not fully benefiting from the conceptual lattice.

After semester 1, most concepts in the lattice were at level 2 which is too general. Several. Similarity in keywords and the fact that a new topic, and that a research topic was introduced each week. Deploying KAPUST another semester a more complex lattice was constructed and comparative research assignments produced a deeper lattice. Students in the following semester had more benefit from the tool, as they searched an existing lattice structure. This illustrated benefits of this approach in an E-Learning environment through knowledge sharing among students across sessions. Pathways in the lattice are automatically captured. When a sufficient number of these is captured, the generated knowledge base is the more and more informative to the students. However, in its early development the knowledge base may not be as informative. The interface allows the students the ability to comment on any webpage they visit and these comments are made available to the whole student body. These comments may offer insight and save

students time, of course these comments may also mislead; in the early stages of the evolution knowledge base, these comments may supersede the utility of the knowledge base to the students. As the knowledge base evolves, it better reflects the holistic view the group and it becomes more accurate, the comments become less important. The knowledge base evolves as the same related set of webpages is traced (websurfed) more than once by more than one student and the same webpages within the same set are visited by multiple students as well. The evolution and the completion of the knowledge base is topic related, that is it may become mature for one topic but highly inaccurate for another.

Formal Concept Analysis generated conceptual structures provide to the students a user friendly natural presentation of the semantic web evolved. The conceptual lattice structures the data from the most general to the most specific concept. It relates concepts to each other based on their intents and extents, thus providing a means for creating new data that was not directly perceived from the user trails. Moreover, querying the lattice is an easy task and it can vary between approaches to make the most out of the structure. For instance, in our method, if a user is searching for a certain concept, we provide him with a categorized result formed of the upper and lower levels of the concept itself in order to gain more insights about the extents constituting the concept.

In its current form, KAPUST provided an E-learning environment which in turn provided the students and the professor involved means of sharing information and experiences without requiring any paper work. Even during the early stage of evolution of the semantic web for the class, the tool still contributed to the learning process. It assisted in collecting web pages under the domain of IT, public administration, and political science and created a structure out of these web pages which accelerated the learning of the class. At the pedagogical level use of KAPUST merges the reading and writing processes; and a whole new dimension of implicit discourse between students evolves on top of the original documents that lend themselves to new multiple reading pathways, and non-hierarchical and polyarchic structures without ruining the integrity of any original research texts.

Pedagogical reflections on KAPUST

Whilst this paper is about the development and use of an innovative web browser, it is an instructive validation to do an educational exploration of learning using this browser as this browsing activity contributes to student learning within the EU learning environment.

The students were set a sequence of comprehension assignments on the topic of IT policies in public administration where the lecturer in charge made available to the students a collection of hypertext body of text based on experts resources. In each of their assignments, they had to browse this body of text to produce written answers for their assignments. This browsing activity is key activity in the environment of synthesis of their assignments. In this process, students collectively learnt and produced KAPUST knowledge base and in some instances leaving useful notes on webpages for their colleagues. Biggs (Biggs 1999) distinguishes between problem based learning (required for maths like subjects) and consumption of large body of information (e.g. in legal studies). Using Biggs classification of university courses (Biggs 1999), KAPUST is a tool which is not recommended for problem based courses it is more suited for courses requiring synthesis of large body of information where the lecturer is able to design assignments and unleash students as a group on the text body using KAPUST. Indeed KAPUST can be seen as an Education Technology (ET) tool with a purpose to provide low level interactivity between students. It enhances presentation of material and overcomes a problem highlighted by Biggs (Biggs 1999) that students often have problems zooming in on the relevant material on the web. Biggs urges lecturers to ensure that students are taught skills to discriminate between the useful and so useful on the web. He suggests training students to be selective and use keywords that cross-classify. These problems are hugely reduced with KAPUST, as the success of keywords is supported by the effort of the whole class and the relationships between the keywords is automatically discovered by KAPUST.

For some assignments the students were required to do comparative analysis between various approaches. By more efficient and more effective searching the students were able to explore deeper and broader in the body of hypertext made available to them more than otherwise possible given their time constraints. Alternatively, for the same time, KAPUST may make it possible for students (as a group) to delve into more difficult text e.g. research papers, simply because they can do it together as a group. This is clearly an advantage to enrich and break the monotonicity of the presentation large lecture classes (Biggs 1999), which is a well cited reason for computer-based learning (Biggs 1999; Barker 2004).

KAPUST is not a typical online discussion which is fundamentally interactive, student -- student, teacher -- students. Using KAPUST the students rather browse the holistic view of the pathways as the

semantic web emerges. They are able to influence its evolution and they are encouraged to allow themselves to be influenced by it. In comparison with a direct online discussion, the result here is slower and perhaps is more inviting for deeper reflections. The recommendations of KAPUST, based on the pathways of all students, is to be taken with a grain of salt by students and students are encouraged to its evolution. As this indirect form of social navigation simulated with KAPUST does not involve direct communication between students, it raises a number of issues:

- Confirming or assessing the reliability of any single source [considered in isolation] is not possible. Instead, that is possible with KAPUST by comparing various views on the same pages and also having access to the original data perused by the users.
- Good will is helpful, but is not essential. In the student environment, there is no incentive to mislead by any of the participants.
- There are no privacy concerns by students who leave behind useful surfing tracks. Association between individuals and webpages is not required and can be destroyed outside an e-learning environment where privacy is of concern and users should be assured of this anonymity to allay any privacy concerns.

7. Summary and future work

We developed an add-on e-learning tool to work with existing internet browsers. We have borrowed several rules from psychological studies of social navigation on what makes a good navigation system: we considered the integration of the tool with the browser, the presence of a community to share knowledge, and the proxemic space to provide a transparent environment for the users. This has further made our tool easier to use. Our tool requires very little training. Anyone familiar with using the internet browser can directly recognize its functionalities. Our hypothesis in this project has been that individuals in a given interest group produce surfing trails that are of interest to the whole group. Our software and our experiments have shown that our hypothesis is indeed true. Individuals of our interest group, students studying political science, produced surfing trails that are of interest to the whole group. Students' interactions with the course notes have incrementally evolved into a Semantic Web over two semesters, describing the relationships amongst contents of their course notes and ideas. These dependencies were utilized by our intelligent search engine to yield a more focused search result, for the students themselves to use as they learn. The whole class was collectively learning and evolving as the subject progresses. The resultant conceptual lattice matured in depth and in number of concepts as relationship between the pages and students' notes was automatically discovered by our intelligent software running in the background, ensuring that the technology we used remains non-intrusive and easy to use to non-technical students.

Throughout this research, in characterising the FCA knowledge base (in KAPUST), we take a situated view of knowledge. That is, the context in which knowledge is consumed (by a student) or re-called to surface into consciousness (of a student) and be articulated (by a student), (this context) plays an integral role into its formation and definition. From KAPUST perspective, a student creates a knowledge in reaction to the search context the student is in. This context is firstly defined by at least the state of the knowledge base, the assignment and the student him/herself and secondly by the webpage visited. A knowledge base developed by the body of students will be conceptual description of the hypertext body. In other words, the students are guided to impose structure on the hypertext body. This is suggested as suitable web based activity by (Barker 2004). This also suggests that KAPUST is a tool which is not suitable for problem based subjects rather more to subjects requiring synthesis of large body of information where the lecturer is able to design assignments and unleash students as a group on the text body using KAPUST. This is a common distinction for university courses used by (Biggs 1999; Barker 2004) amongst others.

Our system, KAPUST, is a form of open multi agent system (MAS). The external agents are the browsers (students in our case) and the internal agents are the intelligence building algorithms and the intelligent interface modules (e.g. dictionary). In this light, our framework for enhancing search results in response to users evolves the behavior of the electronic medium (info space) to improve its effectiveness for the users (students in our case), and is in general terms a form of multi agent system evolution. This form of evolution in response to external agents' behavior is of importance for *electronic markets* agent systems e.g. (Esteva 2003). For example, keeping track of clients' preferences in a non-intrusive manner can be used to evolve market access and maintain products lines (service products or otherwise). The non-intrusive nature of our approach solves the problem of getting feedback from customers who may not have any true interest or inclination to give it. This may be added to KAPUST by using incremental techniques

described in (Drake and Beydoun 2000). We are in the process of integrating both of those mechanisms from (Beydoun, Debenham et al. 2004) to identify more prominent evolutionary factors, and devising a measure of trust from (Beydoun, Hoffmann et al. 2005) to evaluate the reliability of the outcome knowledge.

Acknowledgement

I thank Roman Kultchitsky who was excited to follow through with using KAPUST for a second semester and Grace Manasseh for providing technical support for the students.

References

- Barker, P. (2004). Technology in support of learning. Effective Learning and Teaching in Engineering. C. Baillie and I. Moore, Taylor and Francis Group.
- Beydoun, G., J. Debenham, et al. (2004). Integrating Agents Roles Using Messaging Structure. Pacific Rim Multi Agent System Workshop (PRIMA2004), Auckland, Auckland University.
- Beydoun, G., A. Hoffmann (2000). "Monitoring Knowledge Acquisition Instead of Evaluating Knowledge Bases". Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Lecture Notes in Computer Science Volume 1937, 2000, pp 387-402
- Beydoun, G., R. Kultchitsky, et al. (2007). "Evolving Semantic Web with Social Navigation." Expert Systems with Applications **32**: 265-276.
- Biggs, J. (1999). Teaching for quality learning at university. Buckingham, Open University Press.
- Blunt, R. and C. Ahearn (2000). Creating a Virtual Learning Community. The Sixth International Conference on Asynchronous Learning Networks (ALN2000), Maryland.
- Dieberger, A. (1997). "Supporting Social Navigation on the World Wide Web." International Journal of Human Computer Studies **46**(6): 805-825.
- Drake, B. and G. Beydoun (2000). Using Ripple Down Rules to Build Predicates in Situation Semantics. 6th Pacific Rim Knowledge Acquisition Workshop (PKAW2000), Sydney, University of New South Wales.
- Eklund, P. and B. Wormuth (2005). Restructuring Help Systems Using Formal Concept Analysis. ICFCA. B. Ganter and R. Godin, Springer-Verlag: 129-144.
- Esteva, M. (2003). Electronic Institutions: From Specification To Development. Artificial Intelligence Research Institute. Barcelona, UAB - Universitat Autònoma de Barcelona.
- Ganter, B. and R. Wille (1999). Formal Concept Analysis: Mathematical Foundations, Springer-Verlag.
- Horton, W. and K. Horton (2003). E-learning Tools and Technologies, John Wiley and Sons.
- Kim, M. H. and P. Compton (2001). Formal Concept Analysis for Domain-Specific Document Retrieval Systems. 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI 2001), Ottawa, Springer.
- Laus, F. O. (2001). Tracing User Interactions on World-Wide Webpages. Psychologisches Institut III. Germany, Westfälische Wilhelms-Universität.
- Tilley, T., R. Cole, et al. (2005). A Survey of Formal Concept Analysis Support for Software Engineering Activities. Formal Concept Analysis. B. Ganter and e. al., Springer-Verlag: 250-271.
- Wexelblat, A. (1999). History-Based Tools for Navigation. IEEE's 32nd Hawai'i International Conference on System Sciences (HICSS'99), Hawai, IEEE Computer Society Press.
- Wexelblat, A. and P. Maes (1999). Footprints: History-Rich Tools for Information Foraging. Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh.